

FIG. 1



FIG. 2

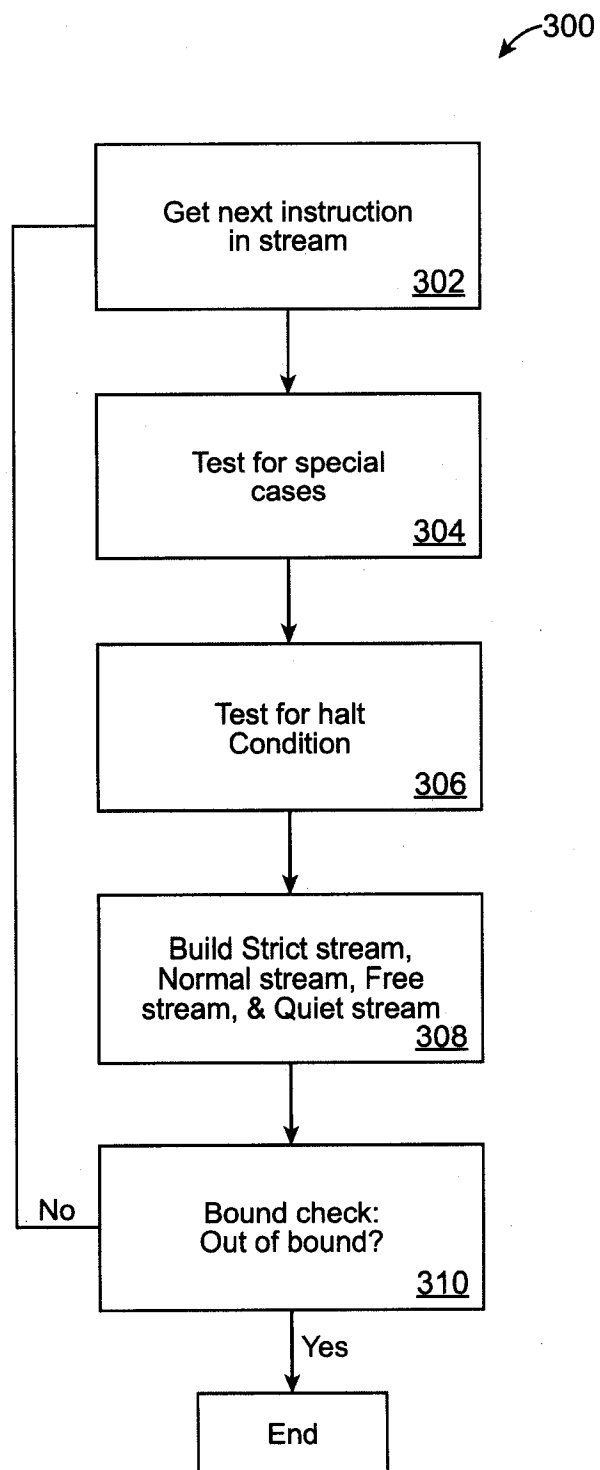


FIG. 3

400

8KB bit-table

101110001001001000010000100000000001
00000100100100010001000100001000101
00010010000010000010001000010001000
10000000100001001000010001001001001
00100010010100010010001000001000001
...
00100100001000100010010111100001010
00010001001000100000010111000010000

FIG. 4

500

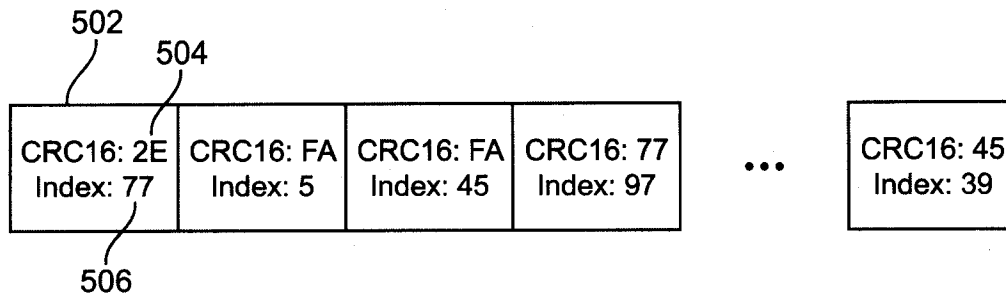


FIG. 5

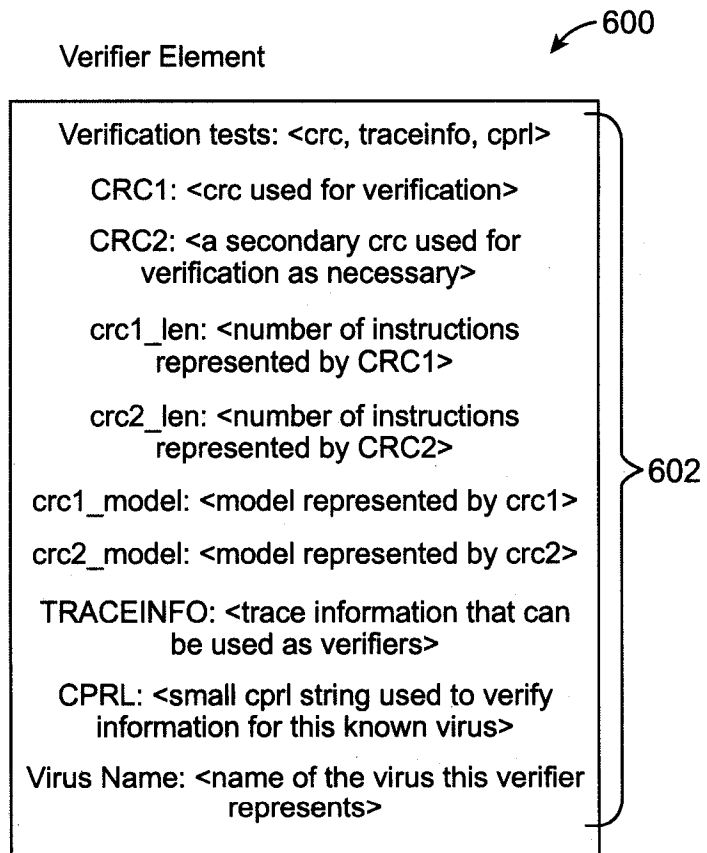


FIG. 6

# REPLACEMENT SHEET

6 / 8

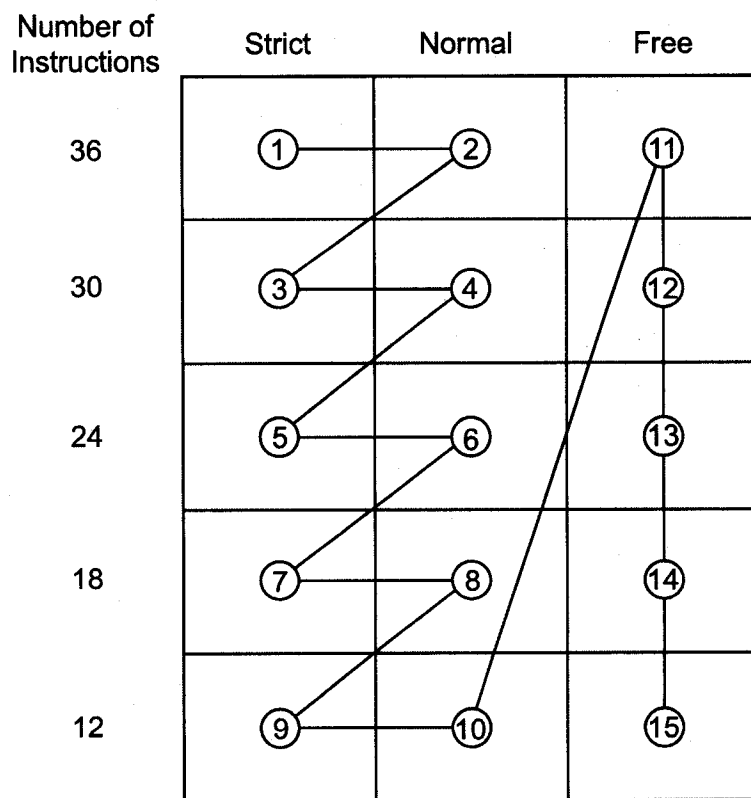


FIG. 7

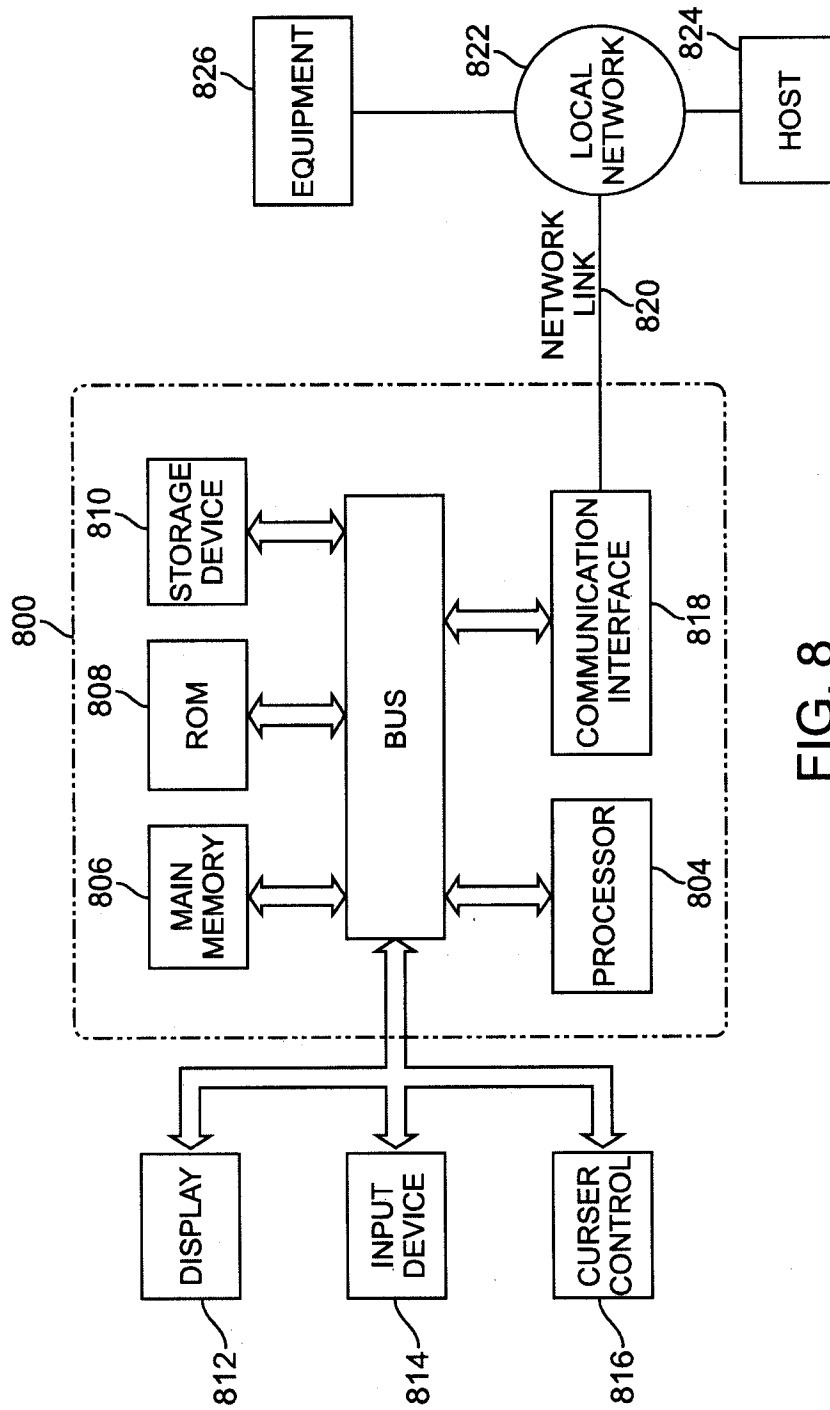


FIG. 8

## REPLACEMENT SHEET

8 / 8

Any two byte CRC contains a value from 0 to 65535. An eight kilobyte table contains 65536 bits. Therefore each bit in the 8k table can therefore represent a value from 0 (the first bit) to 65535 (the last bit). In other words, every value from 0 to 65535 can be represented by a single bit in an 8k table.

The following is an example of an algorithm that can be used to mask a CRC value to a single bit in the table:

C = CRC; B = Byte in table that contains the bit. b = the bit number that represents C. M = mask byte.

Note: Bit values 0 to 7 are represented by location, not by value.

Example:

1000000 = bit 0 set.

0000100 = bit 5 is set.

Algorithm

$b = C \wedge 8$

$B = C / 8$

Set high bit only in M (1000000)

Shift byte right by count b. (if b = 0 then no shift occurs)

Bitwise AND M against B.

If NOT zero then match.

Pseudo Code

```
boolean
MASK (WORD C,          // CRC
      BYTE *TABLE)    // pointer to table
{
    WORD B;            // which byte in table
    BYTE b;            // bit count
    BYTE M;            // bit mask
    BYTE *N;           // pointer to byte in table

    M = 0x8000;        // initialize to binary 10000000

    B = C / 8;         // number of byte in table
    b = C ^ 8;         // remainder of C / 8
    N = TABLE + B;    // point to byte in table
    M = M >> b;        // shift right 0 to 7 bits

    if (M AND *N)      // AND Mask and Byte N in table
        return true;  // bit is set
    return false;      // bit is not set.
}
```

FIG. 9